




HowTo fstab

السلام عليكم ورحمة الله وبركاته

Seven_Eleven

لينكساوي

موضوعنا اليوم عن ملف ال fstab وماهى هى الفوائد التى يمكن أن يقدمها لنا خلال تعاملنا اليومى مع أنظمة ال Unix-Like .

مقدمة :

ملف الإعدادات fstab هو أحد الملفات المهمة على اى توزيعة لينوكس, ويستخدم الملف لتمرير إعداداته إلى الكيرنل للعمل على انظمة الملفات المختلفة مثل reiserFs , ext3 , ext2 إلخ من تلك الأنواع بالإضافة إلى ربط أقسام الهارددديسك أو ال(Partitions) بأنواع نظم ملفات مختلفة أو Filesystems التى تحملها تلك الأقسام فيما يسمى بعملية ال mount .

يوجد ملف ال fstab على المسار /etc/fstab ويحتوى الملف كما ذكرنا سابقا على اعدادات خاصة يتم تمريرها إلى الكيرنل لمعرفة انوع الأقسام المعنية من الهارددديسك وأنواع ملفات النظام الخاصة بها , كما يحتوى أيضا على بعض الخيارات التى يمكن وضعها على تلك الأقسام وسنستعرض ذلك لاحقا.

ولكن ما الفائدة الجوهرية للملف fstab ؟

تكمن الفائدة الجوهرية للملف لأصحاب الأجهزة ذات الإقلاع الثنائى أو فيما يعرف بال dual-boot systems حيث يوجد أكثر من نظام تشغيل , ولكل نظام تشغيل أقسام خاصة من الهارددديسك ولكل قسم نظام ملفات أو filesystem , ولسبب ما يريد المستخدم مشاركة هذه الأقسام مع نظام تشغيل آخر فيتم اللجوء إلى عملية الضم أو فيما يعرف بعملية ال mount وبعد الإنتهاء من هذه العملية نقوم بحفظ إعدادات هذه العملية فى الملف fstab لضمان استمرار عملية ال mount بشكل تلقائى فى حال عمل إعادة تشغيل للجهاز .

بعد أن انتهينا من مقدمة عن الملف أصبح من الواضح أنه توجد علاقة وطيدة بين الأمر mount او عملية ال mount بشكل عام وبين الملف fstab ويتضح من ذلك عدة نقاط :

* الخيارات التى يتم اضافتها مع الأمر mount وإضافتها إلى الملف fstab متشابهة .

* فى حالة عدم وجود قسم من أقسام الهارد أو ال partition فى ملف ال fstab فإن الوحيد القادر على اضافة ذلك القسم هو المستخدم الجذر أو ال root .

* نظم الملفات الخاصة بنظام التشغيل ويندوز و المدعمة تماما ك read و write هى نظم ال fat وال fat32 , أما نظام الملفات NTFS فحتى الآن مدعم بصيغة القراءة فقط أو ال read وذلك من خلال الكيرنل ولكن فى الإصدارات الحديثة من الكيرنل تم إضافة module الكتابة فى اعدادات الكيرنل ولكن حتى الآن ما زال بشكل

تجريبى ، أما لتدعيم القراءة والكتابة على نظام الملفات NTFS فيتم اللجوء إلى استخدام هذا ال driver وهو ntfs-3g.

* نظم الملفات MSDOS , VFAT, NTFS مدعومة بشكل تلقائى فى اعدادات الكيرنل ك modules وفى حالة عدم توافر أين منها يمكن إضافتها لاحقا باستخدام الأمر modprobe وذلك بالصيغة التالية :

كود:

```
modprobe msdos, modprobe vfat and modprobe ntfs.
```

عملية ال mount :

بشكل سريع سأتطرق إلى كيفية عمل ال mount حتى يكون الموضوع متناسق الفقرات ولا تحدث بلبلة لدى القارىء . كما ذكرت سابقا ان عملية ال mount الهدف منها هو ضم أقسام الهارد الغير مرئية بالنسبة للمستخدم لكى تكون متاحة ومرئية بالنسبة له ، بمعنى يوجد لديك نظامان تشغيل أحدهما نظام التشغيل ويندوز والآخر نظام التشغيل لينوكس والآن انت تعمل على نظام التشغيل لينوكس ولديك على الهارديسك الخاص بك أربعة بارتشنات مختلفة اثنان منهم بنظام ملفات FAT32 وواحد بنظام ملفات NTFS والآخر بنظام ملفات EXT3 ، وعندما قمت بالدخول إلى نظام التشغيل لينوكس حدث شىء غريب ألا وهو أن الأقسام الخاصة بنظامىّ الملفات FAT32 و NTFS لا يمكنك قراءة أين منهما فما العمل ؟

هل اختفت الأقسام بلا رجعة ؟ هل حدث خطأ ما ؟؟ الإجابة لا ...لا تقلق فكل ما فى الأمر أن نظام التشغيل لينوكس بشكل تلقائى لا يرى إلا القسم الذى تم تثبيته فيه ولكى تتمكن من العمل على باقى الأقسام لابد من عملية ال mount التى ذكرناها سابقا ولكن كيف لنا أن نقوم بهذه العملية ؟؟

أولا وبشكل بسيط جدا لابد من التعرف على تلك الأقسام وأين توجد بمعنى ، يتعامل نظام التشغيل لينوكس أقسام ال hard drives بشكل مختلف تماما عن نظام التشغيل ويندوز بمعنى على نظام التشغيل ويندوز أقسام الهارد تحمل الحروف التالية C , D , E , F , G , H , I وهكذا ولكن نظام التشغيل لينوكس يتعامل مع الأقسام بشكل مختلف فمسمياته بالنسبة لل hard disks من نوع ATA تكون هكذا hda1 , hda5 , hda6 , hda7 , hda8 وذلك إذا كان الهارديسك على توصيلة primary master أما إذا كان الهارد على توصيلة primary slave فيكون شكل المسميات هكذا hdb1 , hdb5 , hdb6 , hdb7 .

لكن لاحظ التالى فى ترتيب الأقسام ذكرت بالأعلى أن البارتشن رقم واحد يحمل الإسم hda1 وهو المقابل فى ويندوز للبارتشن الذى يحمل الحرف c ولكن لماذا البارتشن الثانى يحمل الإسم hda5 بدلا من hda2 ؟؟ خطأ مطبعى أليس كذلك؟؟

مهلا ليس كذلك , فى لينوكس ذكرت سابقا أن يتعامل فى مسميات أقسام الهارديسك بشكل مختلف وقد ذكرنا ذلك فعلا فقلنا البارتشن الأول الذى يحمل الإسم hda1 مقابل للبارتشن الذى يحمل الإسم c على ويندوز والبارتشن الذى يحمل الإسم hda5 مقابل للبارتشن الذى يحمل الإسم d على ويندوز وهكذا دواليك ولكن المحور الأساسى للمشكلة هنا تسمية البارتشن الثانى على لينوكس بإسم hda5 بدلا من hda2 فكيف ذلك ؟؟

المُسمى hda5 ظهر نتيجة أن أقصى عدد لل primary partitions على أى هارديسك يكون 4 أقسام فقط ، ولو افترضنا أنه يوجد لديك القسم c على ويندوز وهو primary partition فطبيعى سيحمل الإسم hda1 على لينوكس ، بعد ذلك يوجد لديك قسم تحت اسم extended والذى يحتوى على ال logical partitions والقسم extended فى ذاته يكون من النوع primary ويحجز الأرقام من 2 إلى 4 وبالتالى أول بارتشن logical تحت القسم extended سيحمل الرقم 5 وذلك فعلا ما يحدث مع لينوكس .

بعد أن استعرضنا نقطة مسميات أقسام الهارددیسك على لينوكس أو ال partitions ننتقل الآن إلى كيفية عرض تلك الأقسام وذلك يكون من خلال الامر التالى (فى وضعية ال root):

كود:

```
$fdisk -l

  Device Boot      Start         End      Blocks    Id  System
/dev/hda1              1          388     2933248+  83   Linux
/dev/hda2            389          557     1277640    5   Extended
/dev/hda5            389          557     1277608+   b   W95 FAT32
```

كما رأينا فى المثال السابق الإسم hda2 محجوز للقسم extended وتلى ذلك القسم الإسم hda5 وهو أول logical partition تحت القسم extended كما ذكرنا بأعلى .

ولكن كيف أميز البارتشنات أو أقسام الهارددیسك والتى تم عمل mount لها من التى لم يتم عمل mount لها ؟

الاجابة بسيطة قم بفتح الطرفية أو ال shell لديك وقم بكتابة الأمر التالى :

كود:

```
$mount

/dev/hda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
procbususb on /proc/bus/usb type usbfs (rw)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
nfsd on /proc/fs/nfsd type nfsd (rw)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

من الواضح ان البارتشن الذى يحمل الإسم hda5 لم يتم عمل mount له والآن نريد عمل mount له فكيف تتم تلك العملية ؟

الموضوع بسيط جدا لتنفيذ عملية ال mount لابد من إنشاء نقطة للضم أو mount point بمعنى لابد من وجود شىء ما لربط البارتشن المُراد عمل ال mount له مع المكان الأصلى للبارتشن ، وفى المثال السابق المكان الأصلى للبارتشن فى المسار /dev/hda5 ولكى نستطيع تنفيذ عملية ال mount نقوم بإنشاء مجلد آخر تحت اى مسار على النظام لربط البارتشن بنقطة الضم او ال mount point .

يوجد مساران على اى توزيعة لينوكس من المفضل إنشاء نقطة ال mount بهما ألا وهما المسار /mnt والمسار /media .

الآن نقوم بإنشاء نقطة ضم او mount point فى أى من المسارين السابق ذكرهما (تنفيذ الأوامر وأنت root) :

كود:

```
$mkdir /mnt/xxxxx
```

حيث xxxxx اى إسم يخطر ببالك لا توجد قيود على ذلك الإسم مطلقا .

الآن نقوم بتنفيذ أمر ال mount للبارتشن hda5 وهو من نوع نظام ملفات fat32 ويكون ذلك من خلال الأمر التالى :

كود:

```
$mount -t vfat /dev/hda5 /mnt/xxxxx
```

وطبيعى الأمور تسرى على أى نظام ملفات آخر فلو أردنا عمل mount لبارتشن من نوع نظام ملفات ntfs نستبدل فقط كلمة vfat ب ntfs وهكذا مع أى نظام ملفات آخر .

انتهينا من الجزء الأول من الموضوع والآن إلى الجزء الثانى

الملف /etc/fstab :

بعد تلك المقدمة نتطرق الآن إلى الموضوع الأصلى وهو كيفية ضبط اعدادات الملف fstab .

فى البداية نقوم بفتح الملف عن طريق أى محرر نصوص تفضله ، عن نفسى افضل محرر النصوص nano (تنفيذ الأوامر وأنت root) :

كود:

```
$nano /etc/fstab
```

وكمثال لشكل الملف :

كود:

```
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/hda1       /               ext3    defaults,errors=remount-ro 0       1
```

والآن نستعرض محتويات الملف :

كود:

```
fstab Syntax

[file system] [mount point] [type] [options] [dump] [pass]
```

كود:

```
[file system] = Physical location.  هذا القسم من الملف يحدد المسار الأصلى
للبارتشن
```

أمثلة على ذلك :

كود:

```
/dev/hdxy or /dev/sdxy.

x will be a letter starting with a, then b,c,....
y will be a number starting with 1, then 2,3,....
```

كود:

```
[mount point] = هذا القسم من الملف يحدد نقطة الضم و التى تمثل المجلد الذى قمنا بإنشائه
```

كود:

```
In general

   1. /mnt Typically used for fixed hard drives HD/SCSI.
   2. /media Typically used for removable media (CD/DVD/USB/Zip).


Examples:

   1. /mnt/windows
```

```
   2. /mnt/data
   3. /media/usb
```

كود:

```
[type] = هذا القسم من الملف يحتوى على أنواع نظم الملفات المختلفة
```

أمثلة على ذلك :

كود:

```
File types:
Linux file systems: ext2, ext3, jfs, reiserfs, reiser4, xfs, swap.

Windows:
vfat = FAT 32, FAT 16
ntfs= NTFS

CD/DVD/iso: iso9660
```

كود:

```
[options] هذا القسم من الملف يحتوى على الخيارات التى يتم وضعها على البارتشن
```

الخيارت كثيرة والتى توضع فى تحت قسم ال options ولذلك سنستعرض اهمها ومنها :

كود:

```
1-defaults
```

عند وضع كلمة defaults تحت القسم options فإننا نعنى بذلك شيئان هما auto , rw :

كود:

```
rw = تعنى تصريح القراءة والكتابة على البارتشن
auto = تنفيذ عملية المونت عند كل اقلاع للجهاز
```

ملحوظة : الخيار defaults ياعنى عدة خيارت متضمنة داخله ولكنى ذكرت أهم خياران منهم . وتوجد خيارات أخرى متضمنة داخله مثل async , suid , dev , nouser , exec

كود:

```
2-ro : read only filesystem
```

عند وضع ro بدلا من defaults فإننا نغير التصاريح الخاصة بالبارتشن لكى يكون صالح للقراءة فقط أى تصفح محتوياته ولا يمكن الكتابة عليه أى الإضافة إليه او الحذف منه

كود:

```
3-noauto : don't mount at boot
```

هذا الخيار يعنى عدم تنفيذ عملية ال mount اثناء إقلاع الجهاز وكمثال على الخيارين السابقين :

كود:

```
/dev/hda1   /mnt/ntfs   ntfs   ro,noauto
```

لاحظ : عند إضافة اكثر من خيار يتم الفصل بينهم عن طريق ال , او comma .

كود:

```
4-user, users: let users mount and unmount
```

فى البداية إذا تم وضع الخيار user فإننا نعنى بذلك جعل أى مستخدم على النظام بدلا من

المستخدم الجذر أو root من تنفيذ عملية ال mount لأنه كما ذكرنا سابقا عملية ال mount مقتصرة فقط على المستخدم الجذر او root .

ولكن ما فائدة users إذا ؟؟ لاحظ معى إذا تم وضع الخيار user فقط فسيكون بمقدر كل شخص على النظام عمل mount لأى بارتشن ولكن عند عمل unmount للبارتشن أى فك الضم فلن يتمكن أحد من فك الضم أو عملية ال unmount غير المستخدم الذى قام بعملية ال mount ، اما إذا وضعنا users فسيكون لدى اى مستخدم على النظام عمل mount او unmount بدون أى قيود .

مثال على ذلك نفترض أننا قمنا بعمل mount لملف iso على ال cdrom ووضعنا السطر الخاص بعملية ال mount فى الملف fstab كالتالى :

كود:

```
/dev/cdrom   /mnt/cdrom   iso9660   ro,user,noauto
```

فى المثال يستطيع اى مستخدم على النظام وليكن المستخدم muhammad كما قلنا عمل mount لملف ال iso9660 على ال cdrom باستخدام احد الأمرين :

كود:

```
$mount /dev/cdrom
```

or

كود:

```
$mount /mnt/cdrom
```

ولكن عند فك الضم عن ملف ال iso لن يستطيع اى مستخدم آخر غير muhammad (بالطبع غير المستخدم root) على النظام من تنفيذ عملية ال unmount .

اما إذا وضعنا الخيار users فسيكون لدى أى مستخدم القدرة على تنفيذ العمليتين عملية ال mount و عملية ال unmount .

ملحوظة : يوجد خيار ذو صلة بالخيارين السابقين ألا وهو الخيار nouser والذى يعنى قصور أمر ال mount على المستخدم root فقط وهذا الخيار إذا لم يتم وضعه فهو قيمة إفتراضية أو default value.

كود:

```
5-uid,gid: mount as (user,group)
```

يتم استخدام الخياران uid و gid مع أنظمة ملفات (MS-DOS, VFAT, NTFS) حيث عند تنفيذ عملية ال mount ووضع هذان الخياران على أين من الثلاثة فسيكون بمقدر كل من user id و ال group id اللذان تم وضعهما حق رؤية وتصفح محتويات البارتشنات التى تم عمل ال mount لها .

ولكن تستطيع كل من ال uid وال gid التى تخص مستخدم معين على النظام نفذ الامر التالى :

كود:

```
$id <user>
```

حيث بدلا من <user> ضع اسم المستخدم الذى تريد معرفة ال uid وال gid له .

مثال : بعد تنفيذ الأمر :

كود:

```
$id muhammad

uid=1000(muhammad) gid=1000(muhammad)
```

الآن عرفنا كل من ال uid و ال gid الخاص بالمستخدم muhammad و نريد وضع تلك الخيارات فى الملف fstab وتكون على الشكل التالى :

كود:

```
/dev/hda1 /mnt/ntfs ntfs uid=1000,gid=100
```

وبالتالى من خلال المثال السابق سيكون فى مقدور المستخدم muhammad من تصفح محتويات البارتشن hda1 على المسار /mnt/ntfs .

ملحوظة : بدون إضافة أين من الخيارات uid و gid سيكون الأمر قاصر على المستخدم الإفتراضى root او بمعنى uid=0 و gid=0
.

كود:

```
6-exec,noexec: execute binaries or not
```

كود:

```
exec lets you execute binaries that are on that partition,
whereas noexec doesn't let you do that.
noexec might be useful for a partition that contains binaries
you don't want to execute on your system,
or that can't even be executed on your system

exec is the default option, which is a good thing.
Imagine what would happen if you accidentally
used the noexec option with your Linux root partition...
```

كود:

```
هذا القسم من الملف يقوم بعمل نسخة احتياطية من انظمة الملفات [dump]
```

تحت القسم [dump] يتم وضع قيمتين إما أن تكون قيمته ب 0 أو ب 1 وفائدة الأداة dump هى عمل backup لل file system ولكن إلى ماذا تشير كلاً من القيمتين 0 أو 1 ؟

إذا تم وضع القيمة 0 فهذا معناه عدم عمل backup من ال filesystem أما إذا تم وضع القيمة 1 فهذا يشير إلى عمل backup من ال filesystem .

لاحظ : لن تتمكن من عمل backup لل filesystem إلا إذا كانت الأداة dump مُثبتة على التوزيعة الخاصة بك .

كود:

```
هذا القسم من الملف يختص بعمل فحص لنظام الملفات أثناء الإقلاع [pass]
```

تحت القسم [pass] يتم وضع قيمتين أيضا إما أن تكون قيمته ب 0 أو ب 1 وفائدة هذا القسم كما ذكرت فى التعريف عمل fsck أو filesystem check أثناء عملية الإقلاع أو ال boot .

إذا تم وضع القيمة 0 فذلك يشير إلى عدم عمل fsck او filesystem check أثناء الإقلاع أو ال boot ، أما إذا تم وضع القيمة 1 فذلك يشير إلى عمل fsck او filesystem check اثناء عملية الإقلاع أو ال boot .

مازال فى الموضوع بقية ساقوم بإضافتها لاحقا وأرجو عدم وضع ردود شكر بدون جدوى إذا كان هناك أى استفسارات أو إضافات تثرى الموضوع فاهلا ومرحبا

سطام
.: فريق عمل الموسوعة :.
جميل جدا ، ورائع تصدق إنه كان على قائمة : toDO

الحمد لله جا هالموضوع في وقته ..

سؤال :

لما قمت بتنفيذ الأمر :

```
modprobe msdos, modprobe vfat and modprobe ntfs.
```

أعطاني هالرسالة :

```
bash: modprobe: command not found
```

أطالب بتثبيت الموضوع لأنه مهم ومفيد جدا

Seven_Eleven
لينكساوي

السلام عليكم

شكر على التثبيت وإن كان الموضوع لم يكتمل بعد

بالنسبة لسؤال الاخ سطام
اقتباس:

> سؤال :
>
> لما قمت بتنفيذ الأمر :
>
> .modprobe msdos, modprobe vfat and modprobe ntfs
>
> أعطاني هالرسالة :
>
> bash: modprobe: command not found

يبدو أن حزمة module-init-tools لا توجد على التوزيعة الخاصة بك قم بتثبيت الحزمة وستتمكن من إضافة وحذف اى module من خلالها

Black X
.: المراقب العام :.

اوامر الـ modprobe لازم تنفذ كرووت من واقع تجربتي

موضوع مفيد جداً اخي Seven_Eleven

ايضاً اتوقع الموديول ntfs اصبح الآن ntfs-3g في اغلب التوزيعات الحديثة اذا لم اكن مخطئ :)

لي عودة لتكملة القراءة, جزاك الله خير

N!x L0ver
مشرف سابق
اقتباس:

> المشاركة الأصلية كتبت بواسطة سطام
> *جميل جدا ، ورائع تصدق إنه كان على قائمة : toDO*
>
> *الحمد لله جا هالموضوع في وقته ..*
>
> *سؤال :*
>
> *لما قمت بتنفيذ الأمر :*
>
> *modprobe msdos, modprobe vfat and modprobe ntfs.*
>
> *أعطاني هالرسالة :*
>
> *bash: modprobe: command not found*
>
> *أطالب بتثبيت الموضوع لأنه مهم ومفيد جدا*

اخوي اسمحلي

لازم تضيفها على ما أتذكر

```
/etc/rc.conf
```

Seven_Eleven
لينكساوي

صحيح بمناسبة كلام الأخ Black X

الأخ سطام حاول تنفيذ الأوامر وانت root وإذا ظهرت لك الرسالة مرة أخرى فسوف تحتاج لتثبيت الباكج module-init-tools

اقتباس:

ايضاً اتوقع الموديول ntfs اصبح الآن ntfs-3g في اغلب التوزيعات الحديثة اذا لم اكن مخطئ :)

على دييبان النسخة الأخيرة etch لا يوجد ال module بشكل افتراضى ولكن قد يكون موجود على توزيعات اخرى وبنسبة كبيرة منها ubuntu 7.10